EVAT Gamification Module:

Implementation Report

Reinforcement Learning Model

Table of contents:

Section 1: Rationale and Data Preparation

This section outlines the strategic approach to developing a Reinforcement Learning (RL) model for the EVAT Gamification Module. It covers the necessity of creating a synthetic dataset to overcome the absence of real-world user data and details the methodology used to generate a realistic and functional dataset for model training.

1.1. The Need for a Synthetic Dataset

A core objective of the gamification strategy is to use an intelligent system to optimize reward allocation, ensuring that the incentives provided to users effectively drive behaviors that align with EVAT's business goals, such as high-quality data contribution. Reinforcement Learning is the chosen methodology for this task, as it allows an agent to learn an optimal policy for maximizing long-term rewards.

However, a significant challenge in this initial development phase is the complete absence of real user interaction data, as the EVAT gamification features have not yet been launched. To address this "cold start" problem and enable the development and validation of the RL model, a synthetic dataset, simulated_user_actions.json, was generated. This dataset serves as a stand-in for real user behavior, providing the necessary input to train and test the RL environment before it is exposed to actual users.

1.2. Synthetic Dataset Generation

The simulated_user_actions.json dataset was programmatically generated to emulate the patterns and diversity of actions expected from a real user base. The generation process was guided by the following principles to ensure its utility for training:

- **Action Space Definition:** The dataset includes the six core, repeatable user actions defined in the gamification strategy: check_in, report_fault, validate_ai_prediction, discover_new_station_in_black_spot, use_route_planner, and ask_chatbot_question.
- **Weighted Randomization:** To mirror real-world usage patterns, a weighted random generation approach was employed. This ensures that more common, low-effort actions (e.g., check_in, ask_chatbot_question) appear more frequently in the dataset, while high-value but rare actions (e.g., discover_new_station_in_black_spot) are less common. This realism is crucial for training a model that understands the relative frequency of different behaviors.
- **Simulated Inactivity:** To account for sessions where a user might not perform any specific gamified action, a 10% probability of "doing nothing" was introduced into the generation logic, further enhancing the dataset's realism.
- **Dataset Scale:** The final dataset was generated by simulating the behavior of 30-40 unique users, with each user performing approximately 50 actions. This resulted in a dataset of around 2,000 total user actions, providing sufficient data volume for initial training and evaluation.
- **Data Format:** The dataset is stored in a simple and clean JSON format, with each

record representing a single action performed by a user. This format is easily parsable for ingestion into the RL training environment. An example record is structured as follows:


```
{
  "user_id": "user_15",
  "action_type": "check_in"
}
```


Section 2: Reinforcement Learning Environment Design

To train an agent, a custom environment that simulates the EVAT gamification system was developed. This environment, named EVATGamificationEnv, was designed to be compatible with the OpenAI Gymnasium interface, allowing for seamless integration with industry-standard RL libraries like Stable-Baselines3. The environment's design is centered around three key components: the action space, the observation space, and a sophisticated reward shaping mechanism.

2.1. Action and Observation Spaces

- **Action Space:** The environment defines a discrete action space consisting of the six core user actions available in the synthetic dataset. This allows the RL agent to choose one of these six actions at each step of an episode.
- **Observation Space:** The observation space provides the agent with the necessary information about the current state of the environment. It is implemented as a normalized vector that tracks the frequency of each action performed so far within the current episode. This allows the agent to make decisions based on the history of recent actions, which is essential for learning strategies that involve variety and avoiding repetition.

2.2. Reward Shaping Mechanism

The design of the reward function is the most critical aspect of the RL environment, as it directly guides the agent's learning process. The goal is not simply to reward individual actions but to encourage a policy that maximizes long-term engagement and aligns with the gamification strategy's objectives. To achieve this, a multi-faceted reward shaping mechanism was implemented:

- **Base Rewards:** Each of the six actions is assigned a base point value, reflecting its intrinsic value to the EVAT ecosystem. These values are directly aligned with the "ChargePoints" blueprint, with high-value data contributions receiving the highest rewards (e.g., check_in = 10 points, report_fault = 25 points, discover_new_station_in_black_spot = 120 points).
- **Reward Scaling and Clipping:** To ensure numerical stability during the training process, the raw reward values are scaled and clipped. This prevents large reward

values from causing instability in the learning algorithm.

- **Advanced Shaping Rules:** To encourage more complex and desirable behaviors beyond simply choosing the highest-value action, several dynamic rules were added:
    - **Diminishing Returns:** A penalty is applied for repeatedly performing the same action multiple times in a row. This rule directly incentivizes the agent to diversify its behavior.
    - **Exploration Bonus:** An extra reward is given the very first time an action is performed within an episode. This encourages the agent to try out all available actions and discover their potential rewards.
    - **Diversity Bonus:** A small bonus is awarded whenever the agent switches from one action to another. This works in concert with the diminishing returns rule to promote a balanced and varied sequence of actions.

Each training "episode" simulates a short sequence of user interactions, lasting between 30 and 80 steps. The agent's performance in an episode is measured by its final cumulative reward, which reflects how well its chosen policy balanced high-value actions with diversity and exploration.

Section 3: Model Training and Evaluation

This section details the process of training the RL agent using the EVATGamificationEnv environment and the synthetic dataset. It covers the choice of algorithm, the training setup, and a comprehensive analysis of the model's performance and learned behaviors.

3.1. Training Setup

The primary goal of the training was to validate the balance of the gamification design. By observing the policy learned by the RL agent, we can determine if the reward structure is robust or if it encourages trivial strategies (e.g., spamming a single action). A successful outcome is an agent that learns a balanced strategy, confirming the design's viability before it is deployed to real users.

- **Algorithm:** The Proximal Policy Optimization (PPO) algorithm from the stable-baselines3 library was selected for this task. PPO is a state-of-the-art, policy-gradient algorithm known for its stability, reliability, and strong performance across a wide range of tasks, making it an excellent choice for this application.
- **Environment Configuration:** A vectorized training environment was created to optimize the training process. A separate, non-vectorized environment was configured for evaluation purposes.
- **Evaluation Callback:** To monitor progress and save the best-performing model, an EvalCallback was configured. This callback would pause training every 200 timesteps to run a series of evaluation episodes. It logged the mean reward achieved during these evaluations and automatically saved the model weights that achieved the highest mean reward.

- **Training Duration:** The PPO agent was trained for a total of 20,000 timesteps, providing sufficient opportunity for the agent to explore the environment and converge on an effective policy.

3.2. Performance Analysis: The Reward Curve

The model's learning progress was tracked by plotting the mean episode reward from the evaluation runs against the training timesteps. The resulting "Evaluation Reward Curve" provides a clear visualization of the agent's improvement over time.

The analysis of the reward curve reveals a successful training process:

- **Initial Performance:** At the beginning of the training, the agent's performance was relatively low, achieving an average episode reward of around 13-15.
- **Steady Improvement:** As training progressed, the curve shows a distinct and steady upward trend. The agent consistently discovered better strategies, with the mean reward climbing and eventually surpassing 24 in later evaluation runs.
- **Convergence and Stability:** After approximately 10,000 timesteps, the reward curve begins to flatten into a stable plateau. This is a strong indicator of successful convergence, meaning the agent has found a consistent and high-performing policy and is no longer making significant random changes to its strategy.
- **Increased Episode Length:** Alongside the increase in reward, the average length of evaluation episodes also grew, stabilizing at around 60 steps. This demonstrates that the agent learned not just to maximize immediate rewards but to sustain interaction for longer periods, a key goal for user engagement.

3.3. Behavioral Analysis: The Learned Policy

To understand what the agent learned, its behavior during evaluation episodes was analyzed by plotting the frequency of its chosen actions and the average reward it received for each action.

- **Action Distribution:** The "Agent Action Frequency" chart shows that the trained agent developed a clear preference for actions with high base rewards, such as discover_new_station_in_black_spot and validate_ai_prediction. This confirms that the agent successfully identified the most valuable actions as defined by the reward structure.
- **Behavioral Diversity:** Critically, the agent did not exclusively perform only the highest-reward actions. The chart shows that all six actions were chosen at various points. This demonstrates the effectiveness of the reward shaping rules (diminishing returns and diversity bonus), which successfully encouraged the agent to explore and maintain a balance of different actions rather than adopting a simplistic, repetitive strategy.

Section 4: Conclusion

The development and training of the PPO-based reinforcement learning model for reward optimization represent a successful completion of Task 2. The project achieved its primary objectives and provides a strong foundation for the intelligent management of the EVAT gamification economy.

The key interpretations from this work are:

- **Successful Learning:** The agent demonstrated clear and consistent learning, with its performance steadily improving from a random baseline to a stable, high-reward policy.
- **Gamification Design Validation:** The agent's learned behavior provides a powerful validation of the gamification system's design. The fact that the agent learned a balanced policy—favoring high-value actions while still engaging in a variety of behaviors—indicates that the reward shaping is well-calibrated and robust. It successfully avoids the common pitfall of encouraging trivial, single-action spamming.
- **Model Readiness:** The final saved model, ppo_evat_reward_model, serves as a valuable asset. It is a proof-of-concept that fulfills the goal outlined in the implementation roadmap and can be used as a baseline for dynamically adjusting rewards to maximize user engagement and data contribution once the system is live.

This work successfully mitigates the "cold start" problem by leveraging a synthetic dataset and provides confidence that the designed gamification economy is balanced, sustainable, and ready for the next phase of development.